IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Lyapunov-based Control Barrier Functions for Real-Time Safe Navigation in Three-dimension Complex Environments

Fuwei Zhang[1], Zhiwei Hou[2]

***Abstract*—In the field of safe navigation for mobile robots, control barrier functions (CBFs) have garnered significant attention due to their ability to transform complex safety constraints into real-time solvable optimization problems. In this letter, we propose a novel Lyapunov-based CBF framework. It offers the following key advantages: (1) Using a single Control Lyapunov Function (CLF), this method synthesizes spatially shifted CBFs to construct an expansive safe invariant set in obstacle-dense environments. (2) The framework is capable of incorporating existing approaches for constructing quadratic CLF, making it applicable to a wide range of complex nonlinear systems and enhancing its generality and extensibility. (3) It enables real-time synthesis of CBFs, and ensures safety in large-scale 3D environments through efficient CBF-based quadratic programming (CBF-QP). (4) The method ensures safety while inheriting the stability properties of the CLF, allowing the asymptotic convergence of the system state to equilibrium, thus unifying safety and motion stability. To validate efficacy, we rigorously tested the framework in both simulations and hardware experiments.**

***Index Terms*—Robot Safety, Autonomous Vehicle Navigation, Collision Avoidance, Motion Control**

I. Introduction

ENSURING real-time safety guarantees remains a critical challenge for mobile robots operating in complex and partially observable environments. CBFs have emerged as a powerful framework for synthesizing provably safe controllers by encoding complex safety constraints into computationally tractable optimization problems. CBFs enable robots to dynamically maintain set invariance, ensuring that the system state never violates predefined safe regions.

A. Related Work

This section focuses on challenges and solutions in safe navigation scenarios, reviewing research progress from both CBF synthesis and online optimization perspectives.

Manuscript received: July, 14, 2025; Revised September, 30, 2025; Accepted October, 26, 2025.

This paper was recommended for publication by Editor Cosimo Della Santina upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by National Natural Science Foundation of China under Grant 62203481. (Corresponding author: Zhiwei Hou.)

The authors are with the School of Systems Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China (e-mails: zhangfw6@mail2.sysu.edu.cn; houzhw5@mail.sysu.edu.cn).

Digital Object Identifier (DOI): see top of this page.

CBF synthesis aims to design functions that satisfy safety constraints, thereby ensuring that the state of the system stays within the safe set. For systems with simple dynamics, the CBF can be designed manually or directly synthesized using the polynomial Sum of Squares (SOS) method. For example, the work [1] proposed a CBF synthesis theory based on Lyapunov functions, while the work [2], [3] formulated CBFs by decomposing objective functions into SOS forms and transforming them into semidefinite programming (SDP) problems. However, these methods exhibit limited scalability for high-dimensional nonlinear systems [4].

Learning-based methods for synthesizing CBFs have advanced significantly in recent years. The work [5] synthesizes CBFs through constrained optimization over expert demonstrations, enforcing safety guarantees through loss functions. In the approach of Joint Synthesis with Controllers, the integration of CBFs and control policies is achieved through co-optimization [6]. The work [7] jointly learns CBFs and perception-based controllers in an end-to-end framework, enabling safe navigation using raw sensor observations. The work [8] co-trains CBFs and policies using a unified loss function, embedding safety directly into control actions. Recent works [9], [10] have further validated learning-based CBF frameworks on real-world robotic platforms. However, ensuring reliable transfer of safety guarantees from simulation to reality remains a key research focus in this field.

After the synthesis of a valid CBF, the CBF-QP approach is commonly utilized to generate safety-guaranteed control actions [11], [12], [13]. As a representative safety filter technique, CBF-QP adjusts the nominal input to ensure safety by solving an online quadratic programming problem [14], [15]. For example, it can serve as a safety filter integrated with existing planners (e.g. RRT [16], RRT* [17]). The core advantages of CBF-QP lie in its real-time performance and compatibility with existing control frameworks. Compared to non-linear model predictive control (NMPC) [18], [19], [20], which addresses safety through finite-horizon optimization over a receding window [21], which can be prohibitive for mobile robots with limited on-board computing power [22], [23]. In contrast, CBF-QP transforms safety constraints into convex linear inequalities [24], significantly reducing computational complexity and allowing real-time implementation.

The work [25] proposed a safety filter framework based

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

on multiple Lyapunov-based CBFs. This method exploits the system's translational symmetry to generate multiple CBFs from a single Lyapunov function, synthesizing a large controlled invariant set. However, the method requires an offline CBF synthesis using predefined static environmental maps. Moreover, while the synthesis of an individual CBF is straightforward, the cumulative computational load becomes prohibitive because the synthesis process for each CBF requires solving a QP problem. This scalability limitation fundamentally restricts deployment in large-scale environments.

B. Contributions

To overcome the limitations of existing methods, we propose a computationally efficient Lyapunov-based CBF framework, which has the following key contributions.

- **Unified Safety and Stability with Enhanced Generality:** The method synthesizes spatially shifted CBFs from a single CLF, thus unifying safety and motion stability within a coherent framework. The framework is capable of incorporating existing approaches for constructing quadratic CLFs, enhancing its generality and extensibility across a wide range of complex nonlinear systems.
- **Computational Scalability in Large-Scale Environments:** By synthesizing CBFs online within a local neighborhood of the current state and the surrounding environment, the method decouples computational cost from environmental scale, enabling safe navigation in large-scale 3D and partially unknown environments.
- **Experimental Validation in Complex 3D Environments:** The proposed framework is extensively validated in both simulations and real-world UAV experiments, demonstrating reliable safety and real-time performance under challenging conditions including aggressive inputs and dense obstacles.

The rest of this paper is structured as follows. Section II presents the necessary background, including safety definitions as well as the fundamentals of CBFs and CLFs. Section III details the proposed Lyapunov-based CBF framework. Section IV provides validation through simulations and hardware experiments. Finally, conclusions and future research directions are discussed.

II. PRELIMINARIES

We consider a robot with non-linear control-affine dynamics:

$$\dot{x} = f(x) + g(x)u, \tag{1}$$

where $x \in \mathcal{X} \subset \mathbb{R}^n$ and $u \in U \subset \mathbb{R}^m$ is the state and control input of the system. $f : \mathbb{R}^n \to \mathbb{R}^n$, $g : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ assumed to be Lipschitz continuous.

A. Safety and CBF

Building on this model, we now define the concepts of control invariance and safety.

Definition 2.1 (Control Invariance and Safety [24]): A set $C \subset \mathbb{R}^n$ is control invariant for the system if for every $x_0 \in C$, there exists an admissible control $u(\cdot)$ such that the trajectory $x(t)$ of the system satisfies $x(t) \in C$ for all $t \in [0, \infty]$. Consequently, the system is safe for the predefined safe region C (a region free of obstacles) if C is a control-invariant set.

Our method of ensuring safety requires that the system have translational symmetry, which is commonly present in moving robot systems during navigation tasks. Now, let us provide the definition of translational symmetry:

Definition 2.2 (Translational Symmetry): For system (1), the translational symmetry is satisfied if there exists an embedding matrix $T \in \mathbb{R}^{n\times n_q}$ $(0 < n_q \leq n)$ such that:

$$f(x) = f(x - Tq), \quad g(x) = g(x - Tq) \tag{2}$$

for all $q \in \mathbb{R}^{n_q}$.

The definition indicates that the dynamics are unchanged under translation within the column space of T, denoted as col(T), with q representing the displacement along col(T).

From Definition of safety, safety of control-affine systems requires control invariance of the set $\mathcal{C}$. Consider a closed set $\mathcal{C}$ defined by a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$:

$$\mathcal{C} = \{x \in \mathcal{X} \mid h(x) \geq 0\}, \tag{3}$$

$$\partial\mathcal{C} = \{x \in \mathcal{X} \mid h(x) = 0\}, \tag{4}$$

$$\text{Int}(\mathcal{C}) = \{x \in \mathcal{X} \mid h(x) > 0\}, \tag{5}$$

where $\partial\mathcal{C}$ and Int($\mathcal{C}$) denote its boundary and interior.

Definition 2.3 (CBF [1]): A continuously differentiable function $h(x) : \mathbb{R}^n \to \mathbb{R}$ is a valid CBF for the system (1), if it satisfies that

$$\sup_{u\in\mathcal{U}} \{L_f h(x) + L_g h(x)u\} \geq -\alpha(h(x)) \tag{6}$$

for all $x \in \mathcal{X}$ and all $t \in [0, \infty]$, where $L_f h(x) \triangleq \frac{\partial h(x)}{\partial x} f(x)$, $L_g h(x) \triangleq \frac{\partial h(x)}{\partial x} g(x)$, $\alpha(\cdot)$ is an extended $\mathcal{K}\infty$ function.

Due to the invariance property enforced by CBFs, a safety filter can be synthesized such that if $x(0) \in \mathcal{C}$, then $x(t) \in \mathcal{C}$ $\forall t \geq 0$ under appropriate control [26]. To design safe controls that deviate as little as possible from the nominal input u^{nom}, we formulate a CBF-QP:

$$\begin{aligned} u^*(t, x) = \arg\min_{u\in\mathcal{U}} \|u - u^{\text{nom}}(t)\|^2 \\ \text{s.t.} \quad L_f h(x) + L_g h(x)u \geq -\alpha(h(x)). \end{aligned} \tag{7}$$

B. Stability and CLF

Lyapunov function is a scalar function used to analyze the stability of dynamical systems. It provides a powerful tool in control theory for ensuring system stability without explicitly solving the system's differential equations.

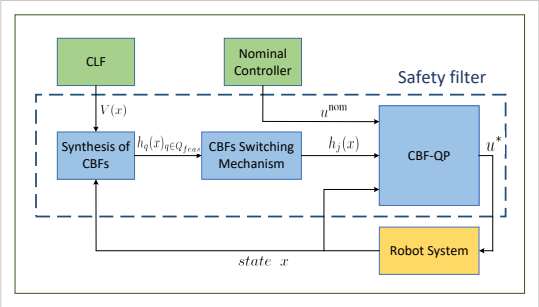


Fig. 1: The framework of safety filter control with Lyapunov-based CBFs.

Definition 2.4 (CLF [27]): A continuously differentiable function $V(x) : \mathbb{R}^n \to \mathbb{R}$ is a valid CLF in the domain $x \in \mathcal{D}$ for the system (1) if it satisfies the following two conditions:

1) Positive definiteness: $V(x) > 0$ for all $x \in \mathcal{D} \setminus \{0\}$ and $V(0) = 0$.
2) There exists an extended $\mathcal{K}\infty$ function $\beta(\cdot)$ such that for all $x \in \mathcal{D} \setminus \{0\}$:

$$\inf_{u\in U} \left\{ \frac{\partial V}{\partial x} \left(f(x) + g(x)u \right) \right\} \leq -\beta(V(x)). \tag{8}$$

If such a CLF $V(x)$ exists, then any state $x \in \mathcal{D}$ can be asymptotically stabilized to the origin. The domain $\mathcal{D}$ represents the region of attraction within which the system's energy (as measured by $V(x)$) is guaranteed to decay to zero under appropriate control. The work in [25] proposes a method to determine the maximum stable domain $\mathcal{D}$ for a given $V(x)$. The procedure involves three main steps: first, define a candidate domain $\mathcal{D}_b = \{x \in \mathbb{R}^n \mid V(x) \leq b\}$ for some $b > 0$; then use a branch-and-bound optimization technique [28] to verify whether $\mathcal{F}(b) < 0$ (which implies that $\mathcal{D}_b$ is a valid stable domain); finally, find the largest b such that $\mathcal{F}(b) < 0$ holds, thus obtaining the maximum stable domain $\mathcal{D}$.

$$\begin{aligned} \mathcal{F}(b) = \max_{x\in\mathbb{R}^n} \min & \left\{ \frac{\partial V}{\partial x}(f(x) + g(x)u) : u \in U \right\} \\ \text{s.t. } & V(x) \leq b. \end{aligned} \tag{9}$$

III. FRAMEWORK OF SAFETY FILTER WITH LYAPUNOV-BASED CBFS

The framework of safety filter with Lyapunov-based CBFs is shown in Fig. 1, which consists of two core modules.

A. Synthesis of Lyapunov-Based CBFs

This section details the systematic procedure for Lyapunov-based CBFs using a single CLF and the system's translational symmetry.

1) Lyapunov-based CBFs: Given a CLF $V(x)$, the translational symmetry of the system allows multiple CBFs to be generated by spatial shifting. For each $q \in Q$, a CBF is synthesized, where Q is a set of points in the free space:

$$h_q(x) = D(q) - V(x - Tq), \tag{10}$$

where $D(q)$, for a fixed q, is a constant satisfying $0 < D(q) < b$. To avoid obstacles, $D(q)$ must satisfy:

$$D(q) = \min \left(b, \min_{q_{\text{obs}}\in Q_{\text{obs}}} m(\delta q) \right) - \epsilon, \tag{11}$$

where q_{obs} denotes the center coordinates of the square obstacles, $\delta q = q_{\text{obs}} - q$, $m(\delta q)$ is the safety margin calculated by (16), and a small positive ϵ is introduced to ensure the strictness of the inequality.

2) Computing $D(q)$: We need to determine q first, and then by matrix multiplication to generate $D(q)$:

1) **Feasible q Generation**: When $x \in \mathcal{C}$, it implies $h_q(x) > 0$. According to (10), this yields $V(x-Tq) < b$. Through that we can define the feasible set of q, the proof see Appendix B:

$$Q_{\text{feas}} = \{q \in \mathbb{R}^{n_q}\} \mid V(x - Tq) < b \wedge q \notin Q_{\text{obs}}\}. \tag{12}$$

We select sampling points q along each dimension $i = 1, \ldots, n_q$ using a constant spatial step Δq, which balances the computational load against the accuracy of the safe set approximation. This process yields a finite set of sampled points, denoted $Q_{\text{finite}} \subset Q_{\text{feas}}$.
2) **collision-free margin**: For each obstacle centered at q_{obs}, compute the collision-free margin:

$$p_i = \begin{cases} \|\delta q_i\| - d, & \text{if } \|\delta q_i\| > d, \\ 0, & \text{else.} \end{cases} \tag{13}$$

where $d = (\frac{d_{\text{obs}}}{2} + r)$, d_{obs} is the obstacle size, r is the safety radius of the moving robot.
3) **Safety Margin**: Decompose the state $x = [x_a, x_b]\top$, where $x_a = p$ corresponds to translation symmetric coordinates (e.g. positions). For quadratic CLF $V(x) = 0.5x^\top Px$, partition P as:

$$P = \begin{bmatrix} P_{aa} & P_{ab} \\ P_{ab}^\top & P_{bb} \end{bmatrix}. \tag{14}$$

Give x_a , P is a positive definite matrix, and its quadratic form is a strictly convex function. So the minimizer x_b of V can be obtained by the derivative method:

$$x_b^* = -P_{bb}^{-1} P_{ab}^\top x_a, \tag{15}$$

yielding:

$$m(\delta q) = 0.5x_a^\top \left(P_{aa} - P_{ab} P_{bb}^{-1} P_{ab}^\top \right) x_a. \tag{16}$$

We can synthesize valid CBFs through the above procedures. For the proof, see Appendix A.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE I: Experiment Parameters

(a) Planar Two-Rotor

Parameter	Symbol	Value
Switch period	δt	0.01 s
Map size	$size_{\text{map}}$	25 m × 25 m
Grid size	d	0.5 m
Obstacle size	d_{obs}	2.0 m
Q_step	Δq	0.2
CLF bound	b	4.0
CBF decay rate	γ	1.0
Robot radius	r	0.2 m

(b) 3D UAV (simulation)

Parameter	Symbol	Value
Switch period	δt	0.01 s
Map size	$size_{\text{map}}$	30 m×30 m×30 m
Cube size	d	0.5 m
Obstacle size	d_{obs}	2.0 m
Q_step	Δq	0.2
CLF bound	b	1.5
CBF decay rate	γ	1.0
Robot radius	r	0.2 m

(c) 3D UAV (hardware)

Parameter	Symbol	Value
Switch period	δt	0.01 s
Map size	$size_{\text{map}}$	4 m×4 m×4 m
Cube size	d	0.2 m
Obstacle size	d_{obs}	0.38 m
Q_step	Δq	0.2
CLF bound	b	2.0
CBF decay rate	γ	1.0
Robot radius	r	0.2 m

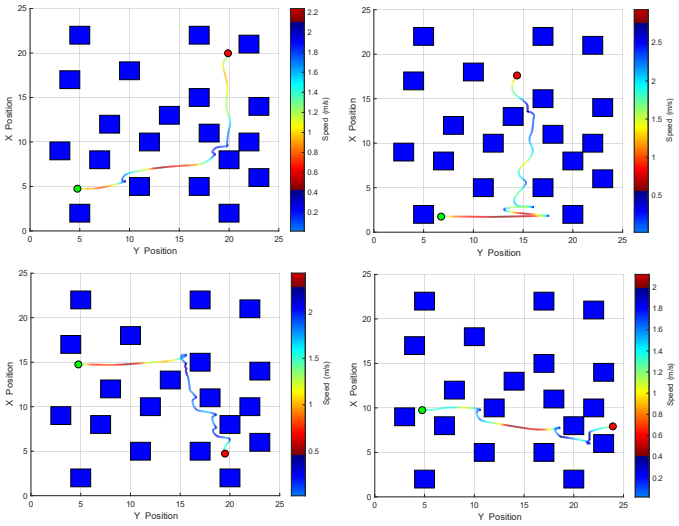


Fig. 2: Simulation results of the Planar Two-Rotor System for various paths. Blue squares represent the obstacle region, and the color variations along the trajectory indicate the robot's speed.

B. CBFs Switching and QP Optimization

This framework ensures persistent safety via dynamically activated CBFs. The composite safe set is as follows:

$$\mathcal{C} = \bigcup_{q \in Q_{\text{finite}}} \{x \mid h_q(x) \geq 0\}, \tag{17}$$

where each $h_q(x)$ is synthesized from a single CLF. Compared to single-CBF methods, our method offers the following:

- **Enhanced Adaptability**: Autonomous switching based on real-time $h_q(x)$ values.
- **Reduced Conservatism**: Select the least restrictive CBF via $j = \arg\max_{q \in Q_{\text{finite}}} h_q(x)$.

1) CBFs Switching Mechanism: During the switching period δt (lower bounded by $\delta t_{\min} > 0$ to prevent Zeno behavior):

- Evaluate $h_q(x(t))$ for all $q \in Q_{\text{finite}}$
- Update active CBF: $j \leftarrow \arg\max_{q \in Q_{\text{finite}}} h_q(x(t))$

2) Control Synthesis with Safety Guarantee: Solve the real-time QP:

$$\begin{aligned} &\arg\min_{u \in U} \|u - u^{\text{nom}}\|^2 \\ &\text{s.t.} \quad L_f h_j(x) + L_g h_j(x) u \geq -\alpha(h_j(x)). \end{aligned} \tag{18}$$

Remark. Under Lipschitz continuity and $\delta t_{\min}$, the QP admits a solution if $x(0) \in \mathcal{C}$ (Theorem 1 [25]).

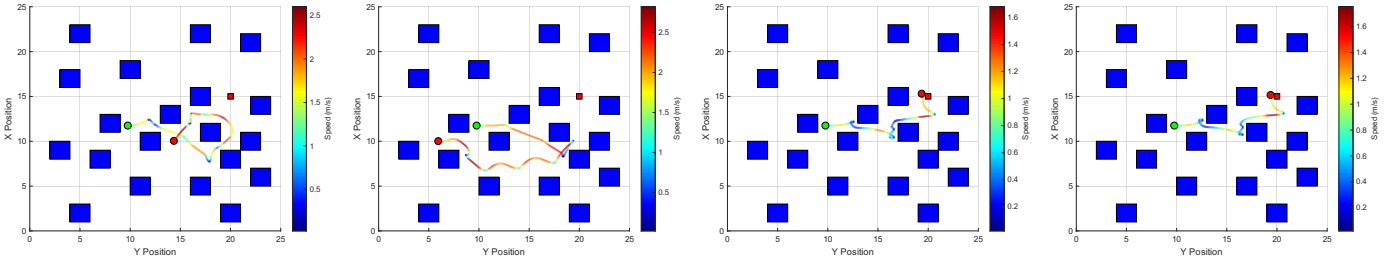
IV. Experiment

In this section, the proposed algorithm is validated through simulation and hardware experiments. For the simulation part, MATLAB software runs on a computer equipped with an Intel(R) Core(TM) i5-1035G1 CPU @1.00GHz. Control performance is verified using a planar two-motor system and a 3D UAV system. In hardware experiments, the simplified nonlinear dynamic model of the translational subsystem is deployed on the DJI Robomaster TT platform for algorithm validation. The detailed values are shown in Table I, and we choose $\alpha(V(x))$ in (18) as the linear function $\gamma V(x)$ (γ is a positive constant).

To validate the proposed framework, two main components are required: a predefined CLF to synthesize the CBFs and an online path planner to generate the nominal input. The choice of quadratic CLF is flexible and can be constructed using any method compatible with the system. To validate the algorithm's capability of integrating both safety filtering and asymptotic stability, we employ a path planner that only provides coarse motion directives. And its nominal control inputs neither guarantee safety nor ensure satisfactory tracking performance. In our experiments, a quadratic Control Lyapunov Function (CLF) $V(x)$, derived from the LQR design, is employed to ensure stability, while the A* algorithm serves as the path planner to generate nominal control inputs.

A. Planar Two-Rotor System

1) System Modeling: The planar two-rotor system follows the dynamics (19) described in the work [25]. The state vector $x = [p_x, v_x, p_y, v_y, \theta, \omega]^\top \in \mathbb{R}^6$ includes position, velocity, tilt angle, and angular velocity. The control input $u = [f_1, f_2]^\top$



(a) baseline CBF-QP (b) APF-CBF-QP (c) OL-CBF-QP (d) Proposed method

Fig. 3: Comparative result of navigation trajectories in a cluttered environment. The green and red circles denote the start and end points of each trajectory respectively, while the red square indicates the goal position.

represents two rotor thrusts constrained by $f_i \in [0, 2]$. The dynamics is given by:

$$\begin{aligned} \dot{p}_x &= v_x, \quad \dot{v}_x = (f_1 + f_2)\sin\theta, \\ \dot{p}_y &= v_y, \quad \dot{v}_y = (f_1 + f_2)\cos\theta - 2, \\ \dot{\theta} &= \omega, \quad \dot{\omega} = f_1 - f_2. \end{aligned} \tag{19}$$

2) CLF Synthesis: We synthesize a quadratic CLF via LQR control:

$$V(x) = \frac{1}{2} x^\top P x, \tag{20}$$

where P is the solution to the algebraic Riccati equation.

3) Safety Filter Implementation: The key implementation steps are summarized in Algorithm 1:

Algorithm 1 Lyapunov-based CBF Safety Filter

1: Measure current state $x(t)$.
2: Generate feasible set Q_{finite} via (12).
3: Compute $D(q)$ for each $q \in Q_{\text{finite}}$ via (11).
4: Calculate $h_q(x)$ for all q via (10).
5: Select $j = \arg\max_{q \in Q_{\text{finite}}} h_q(x)$.
6: Solve QP (18) and apply optimal input u^* with switching period δt.

4) Nominal Input Generation: The nominal input u^{nom} is generated through:

1) The A* algorithm is used to generate a feasible route in a 2D grid map that starts from the current position and ends at the goal position.
2) According to the expected moving direction, the subsequent nominal input u^{nom} is selected:

$$u^{\text{nom}} = \begin{cases} [2, 2]^\top & \text{(upward)} \\ [0, 0]^\top & \text{(downward)} \\ [2, 0]^\top & \text{(rightward)} \\ [0, 2]^\top & \text{(leftward).} \end{cases} \tag{21}$$

If the robot runs away from the route, go to the step 1.

TABLE II: COMPARISON OF COMPUTATIONAL EFFICIENCY AND APPLICABILITY

Scheme	**Offline Time**	**Online Time**	**Dynamic/ 3D Support**
Proposed method	0 ms	30 ± 20 ms	Partial[1] / Yes
OL-CBF-QP method	5 min	65 ms	No / No

To comprehensively evaluate the efficacy of the proposed framework, a comparative analysis was conducted against several established methods: the baseline CBF-QP [24], the Artificial Potential Field-based CBF-QP (APF-CBF-QP [29]), and an off-line Lyapunov-based CBF-QP method [25], herein referred to as the OL-CBF-QP method.

5) Experimental Results: Using our proposed framework, Fig. 2 presents navigation trajectories in a cluttered environment under various start and goal positions. A comparative analysis of the navigation performance achieved by four different methods is provided in Fig. 3. Although both the baseline CBF-QP and APF-CBF-QP methods satisfy safety constraints, they fail to guide the robot to the goal under bang-bang nominal inputs. In contrast, the proposed method ensures both safety and successful goal convergence, owing to its asymptotic stability property. Moreover, the baseline CBF-QP and APF-CBF-QP approaches are difficult to deploy in high-dimensional nonlinear systems. Compared to the OL-CBF-QP method in the 25×25 m $2D$ map with multiple obstacles, which demonstrates comparable navigation performance, our approach achieves significant functional improvements, as summarized in Table II. Specifically, the OL-CBF-QP method requires offline-synthesized CBFs, limiting its applicability in unknown or dynamic environments. Furthermore, it requires considerable computational resources, thereby hindering scalability to large-scale or three-dimensional operational spaces.

[1]Partial dynamic support indicates adaptability to partially known, online-updated maps and aggressive nominal controllers, rather than moving obstacles.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

B. 3D UAV Simulation

The 3D UAV simulation extends the planar system to a three-dimensional space, governed by the 10-state near-hover quadrotor developed in [30]. The full state vector is defined as:

$$x = [p_x, v_x, \theta_x, \omega_x, p_y, v_y, \theta_y, \omega_y, p_z, v_z]^\top \in \mathbb{R}^{10}, \tag{22}$$

where p_x, p_y and p_z denote positions along the x, y and z axes; v_x, v_y, v_z represent the corresponding velocities; θ_x, θ_y are the pitch and roll angles; and ω_x, ω_y are the angular rates. The control input vector consists of:

$$u = [u_x, u_y, u_z]^\top. \tag{23}$$

Here, u_x and u_y denote the desired pitch and roll angles, with constraints $|u_x|, |u_y| \leq 10°$; u_z represents the vertical thrust, subject to the constraint $u_z \in [0.5g, 1.5g]$. The dynamic equations are given by:

$$\dot{p}_x = v_x, \qquad \dot{v}_x = g\tan\theta_x, \tag{24a}$$
$$\dot{\theta}_x = -k_1\theta_x + \omega_x, \qquad \dot{\omega}_x = -k_0\theta_x + n_0 u_x, \tag{24b}$$
$$\dot{p}_y = v_y, \qquad \dot{v}_y = g\tan\theta_y, \tag{24c}$$
$$\dot{\theta}_y = -k_1\theta_y + \omega_y, \qquad \dot{\omega}_y = -k_0\theta_y + n_0 u_y, \tag{24d}$$
$$\dot{p}_z = v_z, \qquad \dot{v}_z = k_T u_z - g. \tag{24e}$$

The parameter values are set as follows: $k_0 = 10$, $k_1 = 8$, $n_0 = 10$, $k_T = 0.91$, and $g = 9.81\text{m/s}^2$.

The nominal input u^{nom} is generated through a 3D extension of the A* path planner. The nominal control inputs are assigned on the basis of the desired movement direction:

$$u^{\text{nom}} = \begin{cases} [\delta_1, \delta_2, 1.5g]^\top & \text{(upward)} \\ [\delta_1, \delta_2, 0.5g]^\top & \text{(downward)} \\ [10°, \delta_2, g/(k_T)]^\top & \text{(forward)} \\ [-10°, \delta_2, g/(k_T)]^\top & \text{(backward)} \\ [\delta_1, 10°, g/(k_T)]^\top & \text{(leftward)} \\ [\delta_1, -10°, g/(k_T)]^\top & \text{(rightward).} \end{cases} \tag{25}$$

where δ_1 and δ_2 are random numbers uniformly distributed within the range $[-5°, 5°]$. u^{nom} exhibits stochastic variability, which means that it will assume different values upon each invocation.

Fig. 4 validates the safe navigation capability of our framework in two cluttered 3D environments, highlighting its robustness and real-time performance.

C. Hardware Experiment

In this subsection, we deploy the simplified nonlinear dynamics model of the translational subsystem on the DJI Robomaster platform for algorithm validation. In the experiment, the iteration time of the proposed algorithm ranged from 15 to 25 ms, meeting the real-time requirements.

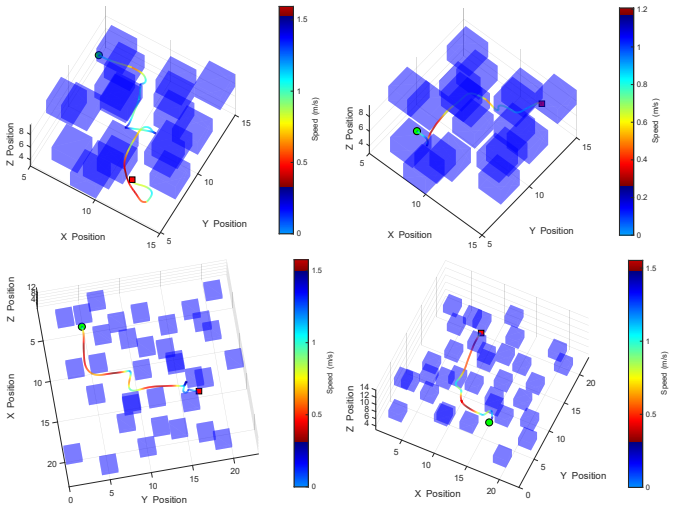


Fig. 4: Simulation results of the 3D-UAV System for various paths. Blue squares represent the obstacle region, and the color variations along the trajectory indicate the robot's speed.

Fig. 5(b) shows the flight scene of the physical experiment. The UAV guided by the safety filter navigates through obstacles safely and reaches the goal position. In Fig. 5(c), the value of h becomes negative during several intervals, possibly due to communication delays. However, this does not result in collisions.

V. CONCLUSION

The paper proposes a Lyapunov-based CBF framework that achieves safe navigation with real-time performance in complex 3D environments. A note on applicability: Although the systems studied in this work did not exhibit higher relative degree issues, the CBFs synthesized by our method can be directly integrated with established techniques such as exponential CBFs [26] to handle such cases should they arise in other applications.

This work opens several promising research directions: Firstly, extending the framework to handle moving obstacles, current experiments focus on static scenarios. A promising direction is to incorporate the obstacle velocity information into the framework. Secondly, the computation of $D(q)$ can be automated by integrating raw sensor inputs with lightweight neural networks, eliminating the need for predefined obstacle representations.

APPENDIX

A. Proof of CBF Validity

For (16), $m(\delta q) = 0.5x_a^\top M x_a$, where M is the Schur complement:

$$M = P_{aa} - P_{ab}P_{bb}^{-1}P_{ab}^\top. \tag{26}$$

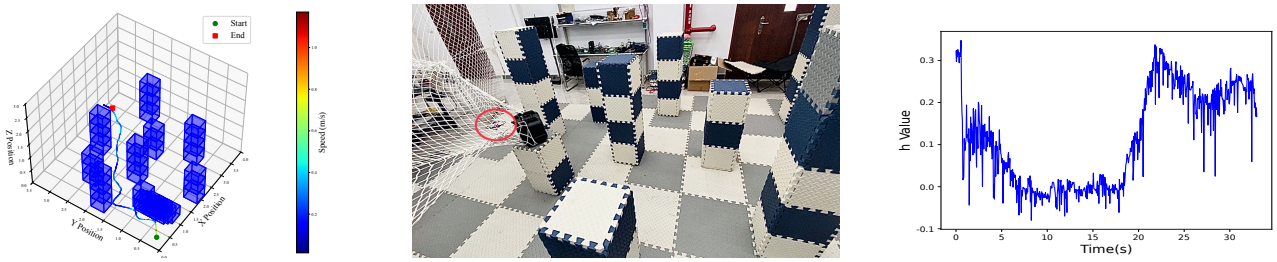


Fig. 5: Real-world experimental results of the 3D-UAV System. For (a), the visualization of experiment, blue squares represent the obstacle region, and the color variations along the trajectory indicate the robot's speed. For (b), the scene of the experiment. For (c), the value of h remains non-negative except for several short intervals.

Since P is a positive definite symmetric matrix, M must be positive definite and symmetric (by Schur complement properties). Thus, it admits a Cholesky decomposition $M = L^\top L$ with a lower triangular matrix L.

According to (16), minimizing $V(x_{\text{obs}} - Tq)$ over variables $x_{b,\text{obs}}$ of x_{obs}, the minimum is determined by:

$$\min_{x_{b,\text{obs}}} V(x_{\text{obs}} - Tq) = 0.5(x_{a,\text{obs}} - q)^\top M(x_{a,\text{obs}} - q), \quad (27)$$

where $x_{a,\text{obs}} - q = \delta q + \delta$ with $\|\delta\|_\infty \le d_{\text{obs}}/2$. The problem reduces to:

$$\min_{\|\delta\|_\infty \le d_{\text{obs}}/2} 0.5(\delta q + \delta)^\top M(\delta q + \delta). \quad (28)$$

Case 1: $|\delta q_i| > d$ ($p_i = |\delta q_i| - d > 0$). By the triangle inequality for absolute values, $|\delta q_i + \delta_i| \ge ||\delta q_i| - |\delta_i||$. Combining $|\delta_i| \le d_{\text{obs}}/2$ and $d = d_{\text{obs}}/2 + r$, we get the following.

$$|\delta q_i + \delta_i| \ge |\delta q_i| - |\delta_i| \ge (d + p_i) - d_{\text{obs}}/2 = p_i + r \ge p_i. \quad (29)$$

Since M is positive definite, $k^\top M k = \|Lk\|^2$ for any vector k. Let $k = \delta q + \delta$, the k-th component of Lk satisfies:

$$\left(\sum_i L_{ki} y_i\right)^2 \ge \left(\sum_i L_{ki} p_i\right)^2, \quad (30)$$

Adding k yields $\|Lk\|^2 \ge \|Lp\|^2 = p^\top M p$.

Case 2: $|\delta q_i| \le d$ ($p_i = 0$). As p corresponds to a point outside the obstacle, $|\delta q_i| \le d$ holds for at most one dimension (the other dimension must have $|\delta q_j| > d$). Assume $p_1 = 0$ for $i = 1$, then $p_2 > 0$ for $i = 2$, and:

$$(\delta q + \delta)^\top M(\delta q + \delta) = y_1^2 M_{11} + 2y_1 y_2 M_{12} + y_2^2 M_{22}. \quad (31)$$

By positive definiteness, $M_{22} > 0$ and $|y_2| \ge p_2$. The cross term satisfies:

$$2|y_1 y_2 M_{12}| \le |y_1|^2 M_{11} + |y_2|^2 M_{22}, \quad (32)$$

thus

$$(\delta q + \delta)^\top M(\delta q + \delta) \ge y_2^2 M_{22} \ge p_2^2 M_{22} = p^\top M p. \quad (33)$$

For all $\|\delta\|_\infty \le d_{\text{obs}}/2$, it follows that:

$$(\delta q + \delta)^\top M(\delta q + \delta) \ge p^\top M p, \quad (34)$$

which implies $m(q_{\text{obs}} - q) \le \min V(x_{\text{obs}} - Tq)$.

Therefore, for any obstacle state $x_{\text{obs}} \in X_{\text{obs}}$, we have:

$$\begin{aligned} h_q(x_{\text{obs}}) &= D(q) - V(x_{\text{obs}} - Tq) && (35)\\ &< m(q_{\text{obs}} - q) - V(x_{\text{obs}} - Tq) && (36)\\ &\le 0. && (37)\end{aligned}$$

Thus, the safe set $\mathcal{C}_q$ excludes all obstacle states.

For any $x \in \mathcal{C}_q$, let $y = x - Tq$. Since $V(y)$ is a CLF, there exists $u^*(y) \in \mathcal{U}$ such that $\dot{V}(y, u^*(y)) \le 0$. Thus:

$$\begin{aligned} \sup_{u\in\mathcal{U}} \{L_f h_q + L_g h_q u\} &= -\inf_{u\in\mathcal{U}} \dot{V}(y,u) \\ &\ge -\dot{V}(y, u^*(y)) \ge 0. \end{aligned} \quad (38)$$

For $h_q(x) \ge 0$:

$$\sup_{u\in\mathcal{U}} \{L_f h_q + L_g h_q u\} + \alpha(h_q) \ge \alpha(h_q) \ge 0. \quad (39)$$

Therefore, $h_q(x)$ satisfies the CBF condition (6) for all $x \in \mathcal{C}_q$, completing the proof of the validity of CBF.

B. Proof of Q is computable

This proof demonstrates that the range of q satisfying $V(x - Tq) \le b$ is computable.

Expanding the constraint with $V(x) = \frac{1}{2} x^\top P x$:

$$\begin{aligned} &(x - Tq)^\top P(x - Tq) \le 2b, \\ &x^\top P x - 2x^\top P T q + q^\top T^\top P T q \le 2b. \end{aligned} \quad (40)$$

Define

$$\begin{aligned} H &\triangleq T^\top P T \in \mathbb{R}^{n_q \times n_q}, && (41)\\ d &\triangleq T^\top P x \in \mathbb{R}^{n_q \times 1}, && (42)\\ c &\triangleq x^\top P x \in \mathbb{R}. && (43)\end{aligned}$$

Then the constraint can be rewritten as

$$q^\top Hq - 2d^\top q + c \leq 2b. \tag{44}$$

Since $P \succ 0$ and T has full column rank, H is symmetric positive definite ($H \succ 0$).

Define $q_0 = H^{-1}d$ and complete the square:

$$\begin{aligned}(q-q_0)^\top H(q-q_0) - q_0^\top Hq_0 + c \leq 2b,\\ (q-q_0)^\top H(q-q_0) \leq 2b - c + d^\top H^{-1}d.\end{aligned} \tag{45}$$

Let $r^2 \triangleq 2b - c + d^\top H^{-1}d$. The feasible set is

$$Q = \left\{ q \in \mathbb{R}^{n_q} \;\middle|\; (q-q_0)^\top H(q-q_0) \leq r^2 \right\}. \tag{46}$$

Since $H \succ 0$, Q is a compact ellipsoid when $r^2 > 0$, indicating that the feasible q values are bounded. For each coordinate i, the extrema $q_i^{\min}$ and $q_i^{\max}$ can be obtained by solving convex quadratic programs with the above constraint, which are efficiently solvable.

Hence, the admissible range of q satisfying $V(x - Tq) \leq b$ is computable.

REFERENCES

[1] P. Wieland and F. Allgöwer, "Constructive safety using control barrier functions," *IFAC Proceedings Volumes*, vol. 40, no. 12, pp. 462–467, 2007, 7th IFAC Symposium on Nonlinear Control Systems. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1474667016355690

[2] H. Dai and F. Permenter, "Convex synthesis and verification of control-lyapunov and barrier functions with input constraints," in *2023 American Control Conference (ACC)*, 2023, pp. 4116–4123.

[3] M. Schneeberger, F. Dorfler, and S. Mastellone, "Sos construction of compatible control lyapunov and barrier functions," *IFAC PAPERSONLINE*, vol. 56, no. 2, pp. 10 428–10 434, 2023, 22nd World Congress of the International Federation of Automatic Control (IFAC), Yokohama, JAPAN, JUL 09-14, 2023.

[4] M. Srinivasan, M. Abate, G. Nilsson, and S. Coogan, "Extent-compatible control barrier functions," *Systems & Control Letters*, vol. 150, p. 104895, 2021.

[5] A. Robey, H. Hu, L. Lindemann, H. Zhang, D. V. Dimarogonas, S. Tu, and N. Matni, "Learning control barrier functions from expert demonstrations," in *2020 59th IEEE Conference on Decision and Control (CDC)*, 2020, pp. 3717–3724.

[6] C. Dawson, S. Gao, and C. Fan, "Safe control with learned certificates: A survey of neural lyapunov, barrier, and contraction methods for robotics and control," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1749–1767, 2023.

[7] C. Dawson, B. Lowenkamp, D. Goff, and C. Fan, "Learning safe, generalizable perception-based hybrid control with certificates," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1904–1911, 2022.

[8] Z. Qin, K. Zhang, Y. Chen, J. Chen, and C. Fan, "Learning safe multi-agent control with decentralized neural barrier certificates," 2021. [Online]. Available: https://arxiv.org/abs/2101.05436

[9] M. De Sa, P. Kotaru, and K. Sreenath, "Point cloud-based control barrier function regression for safe and efficient vision-based control," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 366–372.

[10] S. Keyumarsi, M. W. S. Atman, and A. Gusrialdi, "Lidar-based online control barrier function synthesis for safe navigation in unknown environments," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1043–1050, 2024.

[11] M. Desai and A. Ghaffari, "Clf-cbf based quadratic programs for safe motion control of nonholonomic mobile robots in presence of moving obstacles," in *2022 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2022, pp. 16–21.

[12] M. Tayal, R. Singh, J. Keshavan, and S. Kolathaya, "Control barrier functions in dynamic uavs for kinematic obstacle avoidance: A collision cone approach," in *2024 American Control Conference (ACC)*, 2024, pp. 3722–3727.

[13] J. Liu, M. Li, J. W. Grizzle, and J.-K. Huang, "Clf-cbf constraints for real-time avoidance of multiple obstacles in bipedal locomotion and navigation," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 10 497–10 504.

[14] W. Xiao and C. Belta, "High-order control barrier functions," *IEEE Transactions on Automatic Control*, vol. 67, no. 7, pp. 3655–3662, 2022.

[15] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*, 2019, pp. 3420–3431.

[16] A. Manjunath and Q. Nguyen, "Safe and robust motion planning for dynamic robotics via control barrier functions," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 2122–2128.

[17] T. Kim and D. Panagou, "Visibility-aware rrt* for safety-critical navigation of perception-limited robots in unknown environments," *IEEE Robotics and Automation Letters*, vol. 10, no. 5, pp. 4508–4515, 2025.

[18] N. Piccinelli, F. Vesentini, and R. Muradore, "Mpc based motion planning for mobile robots using velocity obstacle paradigm," in *2023 European Control Conference (ECC)*, 2023, pp. 1–6.

[19] S. Zou, H. Zhang, Y. Lyu, Y. Guo, and G. Ma, "Mpc-based position control for multi-segment pneumatic soft manipulator considering obstacle avoidance," *IEEE Robotics and Automation Letters*, vol. 10, no. 4, pp. 3715–3722, 2025.

[20] J. Liu, J. Yang, J. Mao, T. Zhu, Q. Xie, Y. Li, X. Wang, and S. Li, "Flexible active safety motion control for robotic obstacle avoidance: A cbf-guided mpc approach," *IEEE Robotics and Automation Letters*, vol. 10, no. 3, pp. 2686–2693, 2025.

[21] F. Allgöwer, R. Findeisen, and Z. Nagy, "Nonlinear model predictive control: From theory to application," *JOURNAL OF THE CHINESE INSTITUTE OF CHEMICAL ENGINEERS*, vol. 35, no. 3, pp. 299–315, 5 2004, 2nd Asian Symposium on Process Systems Engineering, Taipei, TAIWAN, DEC 04-06, 2002.

[22] S. Hwang, I. Jang, D. Kim, and H. J. Kim, "Safe motion planning and control for mobile robots: A survey," *INTERNATIONAL JOURNAL OF CONTROL AUTOMATION AND SYSTEMS*, vol. 22, no. 10, pp. 2955–2969, 10 2024.

[23] A. S. Lafmejani, S. Berman, and G. Fainekos, "Nmpc-lbf: Nonlinear mpc with learned barrier function for decentralized safe navigation of multiple robots in unknown environments," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 10 297–10 303.

[24] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[25] I. Jang and H. J. Kim, "Safe control for navigation in cluttered space using multiple lyapunov-based control barrier functions," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2056–2063, 2024.

[26] B. Li, S. Wen, Z. Yan, G. Wen, and T. Huang, "A survey on the control lyapunov function and control barrier function for nonlinear-affine control systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 3, pp. 584–602, 2023.

[27] Z. Artstein, "Stabilization with relaxed controls," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 7, no. 11, pp. 1163–1173, 1983. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0362546X83900494

[28] S. Kousik, B. Zhang, P. Zhao, and R. Vasudevan, "Safe, optimal, real-time trajectory planning with a parallel constrained bernstein algorithm," *IEEE Transactions on Robotics*, vol. 37, no. 3, pp. 815–830, 2021.

[29] A. Singletary, K. Klingebiel, J. Bourne, A. Browning, P. Tokumaru, and A. Ames, "Comparative analysis of control barrier functions and artificial potential fields for obstacle avoidance," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 8129–8136.

[30] P. Bouffard, "On-board model predictive control of a quadrotor helicopter: Design, implementation, and experiments," Master's thesis, University of California, Berkeley, 2012.